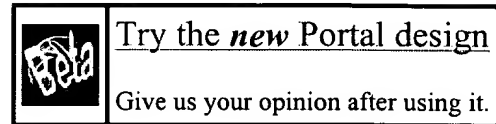




[> home](#) [> about](#) [> feedback](#) [> login](#)

US Patent & Trademark Office



Search Results

Search Results for: **[(thread-specific heaps for multi-threaded programs)]**

Found **7** of **127,944** searched.

Scanned all

Search within Results

60

[> Advanced Search](#)

[> Search Help/Tips](#)

Sort by: [Title](#) [Publication](#) [Publication Date](#) [Score](#) [Binder](#)

Results 1 - 7 of 7 short listing

- 1** Connectivity-based garbage collection 80%

Martin Hirzel , Amer Diwan , Matthew Hertz

ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications October 2003

Volume 38 Issue 11

We introduce a new family of connectivity-based garbage collectors (Cbgc) that are based on potential object-connectivity properties. The key feature of these collectors is that the placement of objects into partitions is determined by performing one of several forms of connectivity analyses on the program. This enables partial garbage collections, as in generational collectors, but without the need for any write barrier. The contributions of this paper are 1) a novel family of garbage c ...
- 2** Beltway: getting around garbage collection gridlock 80%

Stephen M Blackburn , Richard Jones , Kathryn S McKinley , J Eliot B Moss

ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation May 2002

Volume 37 Issue 5

We present the design and implementation of a new garbage collection framework that significantly generalizes existing copying collectors. The *Beltway* framework exploits and separates object age and incrementality. It groups objects in one or more increments on queues called *belts*, collects belts independently, and collects increments on a belt in first-in-first-out order. We show that Beltway configurations, selected by command line options, act and perform the same as semi-space, ...
- 3** Thread-specific heaps for multi-threaded programs 80%

Bjarne Steensgaard

ACM SIGPLAN Notices , Proceedings of the second international symposium on Memory management October 2000

Volume 36 Issue 1

Garbage collection for a multi-threaded program typically involves either stopping all

BEST AVAILABLE COPY